

# STUDI PERBANDINGAN ANTARA ALGORITMA *BIVARIATE MARGINAL DISTRIBUTION* DENGAN ALGORITMA GENETIKA

Chastine Fatichah, Imam Artha Kusuma, Yudhi Purwananto

Jurusan Teknik Informatika, Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember Surabaya

Email: [chf@its-sby.edu](mailto:chf@its-sby.edu); [toshi\\_kusuma@yahoo.com](mailto:toshi_kusuma@yahoo.com); [yudhi@its-sby.edu](mailto:yudhi@its-sby.edu)

**ABSTRAK:** *Bivariate Marginal Distribution Algorithm* merupakan perkembangan lebih lanjut dari *Estimation of Distribution Algorithm*. Algoritma heuristik ini mengenalkan pendekatan baru dalam melakukan rekombinasi untuk membentuk individu baru, yaitu tidak menggunakan proses *crossover* dan mutasi seperti pada *Genetic Algorithm*. *Bivariate Marginal Distribution Algorithm* menggunakan keterkaitan pasangan variabel dalam melakukan rekombinasi untuk membentuk individu baru. Keterkaitan antar variabel tersebut ditemukan selama proses optimasi berlangsung. Aplikasi yang dibuat dalam penelitian ini ditujukan untuk membandingkan kinerja *Genetic Algorithm* sederhana persilangan satu titik dengan *Bivariate Marginal Distribution Algorithm* pada kasus Onemax, Fungsi De Jong F2, dan *Traveling Salesman Problem*. Dari uji coba yang dilakukan, didapat hasil bahwa kinerja dari kedua algoritma tersebut dipengaruhi oleh parameter masing-masing dan juga besar ukuran populasi yang digunakan. Untuk kasus Onemax dengan ukuran masalah yang kecil, *Genetic Algorithm* lebih unggul dalam hal jumlah iterasi yang lebih sedikit dan waktu yang lebih cepat untuk mendapat hasil optimal. Namun, *Bivariate Marginal Distribution Algorithm* lebih unggul dalam hal hasil optimasi pada kasus Onemax dengan ukuran masalah yang lebih besar. Untuk Fungsi De Jong F2, *Genetic Algorithm* lebih unggul dari *Bivariate Marginal Distribution Algorithm* utamanya dalam hal jumlah iterasi dan waktu. Sedangkan untuk kasus *Traveling Salesman Problem*, *Bivariate Marginal Distribution Algorithm* dapat menunjukkan kinerja yang lebih baik dari *Genetic Algorithm* dalam hal hasil optimasi.

**Kata kunci:** *heuristic algorithm, estimation of distribution algorithm, bivariate marginal distribution algorithm, genetic algorithm.*

**ABSTRACT:** *Bivariate Marginal Distribution Algorithm* is extended from *Estimation of Distribution Algorithm*. This heuristic algorithm proposes the new approach for recombination of generate new individual that without crossover and mutation process such as genetic algorithm. *Bivariate Marginal Distribution Algorithm* uses connectivity variable the pair gene for recombination of generate new individual. Connectivity between variable is doing along optimization process. In this research, genetic algorithm performance with one point crossover is compared with *Bivariate Marginal Distribution Algorithm* performance in case Onemax, De Jong F2 function, and *Traveling Salesman Problem*. In this research, experimental results have shown performance the both algorithm is dependence of parameter respectively and also population size that used. For Onemax case with size small problem, *Genetic Algorithm* perform better with small number of iteration and more fast for get optimum result. However, *Bivariate Marginal Distribution Algorithm* perform better of result optimization for case Onemax with huge size problem. For De Jong F2 function, *Genetic Algorithm* perform better from *Bivariate Marginal Distribution Algorithm* of a number of iteration and time. For case *Traveling Salesman Problem*, *Bivariate Marginal Distribution Algorithm* have shown perform better from *Genetic Algorithm* of optimization result.

**Keywords:** *heuristic algorithm, estimation of distribution algorithm, bivariate marginal distribution algorithm, genetic algorithm.*

## PENDAHULUAN

*Genetic Algorithm* (GA) bekerja dengan rangkaian populasi yang berukuran tetap. Dari populasi yang ada, kromosom yang lebih baik akan lebih dipilih dibandingkan dengan kromosom yang buruk. Rangkaian baru dihasilkan dengan menggunakan rekombinasi/penyilangan operator dan mutasi. Rekombinasi populasi menggabungkan informasi yang terkandung di dalam dua kromosom sedangkan mutasi melakukan gangguan pada rangkaian kro-

mosom untuk menjaga keanekaragaman populasi dan mengenalkan informasi baru. Proses ini diketahui dapat menyebabkan gangguan pada kromosom yang menyebabkan algoritma ini berpenampilan buruk pada masalah yang membutuhkan hasil optimum. Hal ini akhirnya menuntun pada pendekatan baru dalam melakukan rekombinasi yang dinamakan *Estimation of Distribution Algorithm* (EDA). EDA pertama kali diperkenalkan dalam lingkup *Evolutionary Computation* oleh Muehlenbein dan Paab (1996) [4].

Perkembangan dari EDA lebih lanjut adalah *Univariate Marginal Distribution Algorithm* (UMDA) (Muehlenbeim, 1998), *Population Based Incremental Learning* (PBIL) (Baluja, 1994) dan *Compact Genetic Algorithm* (cGA) (Harik, 1998). Ketiga algoritma tersebut memberikan hasil yang baik pada kromosom dengan dependensi yang rendah di antara variabel-variabelnya, tetapi tidak memberikan hasil yang baik untuk tingkat ketergantungan yang lebih tinggi. Untuk memecahkan masalah tersebut, diperkenalkanlah *Bivariate Marginal Distribution Algorithm* (BMDA) (Pelikan dan Muehlenbeim, 1999) [1].

Hal yang menarik mengenai pendekatan baru dalam proses rekombinasi ini adalah apakah pendekatan baru tersebut benar-benar dapat menunjukkan nilai lebih dibanding GA. Untuk itu, perlu kiranya untuk membandingkan algoritma dengan pendekatan baru tersebut dengan GA.

## DASAR TEORI

Bagian ini memberikan penjelasan mengenai Algoritma Genetika dan Algoritma Bivariate Marginal Distribution.

### Algoritma Genetika

GA pertama kali dikembangkan oleh John Holland dari Universitas Michigan (1975). John Holland mengatakan bahwa setiap masalah yang berbentuk adaptasi (alami atau buatan) dapat diformulasikan dalam terminologi genetika. GA adalah simulasi dari proses evolusi Darwin dan operasi genetika atas kromosom.

Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin yang dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol.

Populasi awal dibentuk secara acak, sedangkan populasi berikutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah generasi. Pada setiap generasi, kromosom akan melalui proses evaluasi dengan menggunakan alat ukur yang disebut dengan fungsi *fitness*. Nilai *fitness* dari suatu kromosom akan menunjukkan kualitas kromosom dalam populasi tersebut. Generasi berikutnya dikenal dengan istilah anak (*offspring*) yang terbentuk dari gabungan 2 kromosom generasi sekarang yang bertindak sebagai induk (*parent*) dengan menggunakan operator penyilangan (*crossover*).

Selain operator penyilangan, suatu kromosom dapat juga dimodifikasi dengan menggunakan

operator mutasi. Populasi generasi yang baru dibentuk dengan cara menyeleksi nilai *fitness* dari kromosom induk (*parent*) dan nilai *fitness* dari kromosom anak (*offspring*), serta menolak kromosom-kromosom yang lainnya sehingga ukuran populasi (jumlah kromosom dalam suatu populasi) adalah konstan. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.

### Algoritma Bivariate Marginal Distribution

BMDA merupakan pengembangan dari UMDA. Algoritma ini menggunakan keterkaitan pasangan gen untuk mengembangkan UMDA yang sederhana. BMDA merupakan contoh khusus dari Algoritma *Factorization Distribution* (FDA) tapi tanpa adanya pengetahuan spesifik pada taraf awal. Pengetahuan tentang ketergantungan antar gen ditemukan selama proses optimasi itu sendiri.

BMDA dimulai dengan pembentukan populasi awal secara acak dilanjutkan dengan pemilihan himpunan *selected parents* untuk kemudian dihitung nilai *univariate* dan *bivariate marginal frequencies*-nya. Dengan menggunakan nilai *univariate* dan *bivariate marginal frequencies* tersebut, tingkat ketergantungan antar gen atau posisi bit dihitung dengan menggunakan *Pearson's chi-square statistics*.

Tahap selanjutnya adalah pembentukan *dependency graph* dengan menggunakan informasi ketergantungan antar gen yang telah didapat sebelumnya. *Graph* tersebut didefinisikan oleh 3 himpunan  $V$ ,  $E$ , dan  $R$ , yaitu  $G = (V, E, R)$ .  $V$  adalah himpunan dari *vertex-vertex*,  $E \subset V \times V$  adalah himpunan dari *edge-edge*, dan  $R$  adalah himpunan yang berisi satu *vertex* dari tiap komponen yang terkoneksi dari  $G$ .

Dalam pembentukan *dependency graph*, *graph* tidak harus terhubung. Ini berarti bahwa *graph* tidak harus selalu berbentuk *tree*. *Dependency graph* selalu berbentuk *acyclic*. Generasi dari rangkaian baru tidak tergantung pada jumlah dari komponen terhubung dari *dependency graph*.

Dari *dependency graph* yang terbentuk, dibentuklah susunan individu baru. Untuk setiap individu, nilai posisi yang terkandung di himpunan *Root* ( $R$ ) dihasilkan oleh *univariate marginal frequency*. Lalu, jika terdapat posisi  $v$  yang belum dihasilkan dan terhubung pada beberapa posisi  $v'$  (berdasarkan pada himpunan *edge*  $E$ ), nilai posisi akan dibentuk menggunakan probabilitas kondisional untuk posisi  $v$  terhadap posisi  $v'$ .

Individu baru kemudian ditambahkan pada populasi lama dari tempat asal individu-individu tersebut dipilih sebelumnya. Mereka menggantikan beberapa yang lama, biasanya yang terburuk dari mereka,

sehingga jumlah individu-individu dalam populasi tetap konstan.

Dari populasi baru, individu-individu dipilih kembali. Proses ini, dimulai dengan pemilihan individu yang lebih baik dan berakhir dengan penambahan individu-individu baru pada populasi lama, diulang sampai ditemukan kondisi terakhir (*termination criteria*). *Termination criteria* itu menyebabkan algoritma berhenti jika sudah menemukan hasil optimum atau keragaman populasi terlalu rendah. Nilai optimum biasanya tidak diketahui oleh para keturunannya. Inilah alasan mengapa kondisi kedua menjadi hal penting. Saat keragaman menjadi terlalu rendah, hampir semua individu dalam populasi adalah sama. Ini berarti bahwa tidak ada cukup informasi dalam populasi untuk membuat individu-individu baru yang cocok dengan masalah yang lebih baik daripada yang telah ditemukan.

### Masalah Optimasi

Berikut ini penjelasan mengenai masalah optimasi yang digunakan dalam perbandingan kinerja GA dan BMDA.

#### Fungsi Onemax

Onemax merupakan fungsi *linear* sederhana dengan nilai semua bitnya adalah 1. Ini berarti nilai maksimal dari fungsi ini adalah jumlah dari seluruh bit.

$$\text{Persamaan: } f_{\text{onemax}} = \sum_{i=0}^{n-1} x_i \quad (1)$$

di mana  $x_i$  adalah nilai bit pada posisi ke- $i$  pada rangkaian  $x$  dan  $n$  adalah jumlah bit.

#### Fungsi DeJong F2

Fungsi De Jong F2 merupakan fungsi yang sering digunakan dalam uji coba algoritma optimasi pada masalah-masalah numerik. Nama fungsi ini diambil dari nama orang yang membuat fungsi tersebut yang juga seorang pengajar di University of Wisconsin USA.

$$\text{Fungsi: } f(x_1, x_2) = 100 (x_1^2 - x_2)^2 + (1 - x_1)^2 \quad (2)$$

dimana:

$$-2,048 \leq x_1 \leq 2,048$$

$$-2,048 \leq x_2 \leq 2,048$$

Fungsi ini mempunyai dua parameter yaitu  $x_1$  dan  $x_2$  yang akan mencapai nilai minimal yaitu 0 bila masing-masing parameter tersebut bernilai 1.

### Traveling Salesman Problem

*Traveling Salesman Problem* (TSP) merupakan masalah optimasi yang sudah sering digunakan oleh para peneliti karena kesederhanaannya, pentingnya, serta hubungannya dengan masalah kombinatorial.

Konsep TSP sangat sederhana, yaitu seorang salesman harus mengunjungi setiap kota tujuan hanya sekali dan kembali lagi ke kota asal. Dengan adanya jarak/beban biaya antar kota, salesman tersebut harus menempuh rute dengan jarak/beban biaya terkecil.

### HASIL UJI COBA

Berikut ini penjelasan mengenai proses dan hasil uji coba yang dilakukan untuk mengetahui kinerja GA dan BMDA pada penyelesaian masalah optimasi.

#### Skenario Uji Coba

Kinerja GA dan BMDA dipengaruhi oleh parameter masing-masing. Untuk itu, sebelum membandingkan kinerja kedua algoritma tersebut, perlu juga diuji pengaruh parameter terhadap kinerja masing-masing algoritma. GA dan BMDA memiliki parameter yang berbeda. Parameter untuk GA adalah peluang *crossover* ( $P_c$ ) dan peluang mutasi ( $P_m$ ). Sedangkan parameter untuk BMDA adalah jumlah *parents* dan jumlah individu baru. Pengaruh besar populasi juga diuji pada masing-masing algoritma. Perbandingan dilakukan untuk ketiga contoh kasus yang ada dengan menggunakan parameter-parameter terbaik dari uji coba tersebut.

#### Uji Coba Pengaruh Parameter GA pada Kasus Onemax

Uji coba pertama dilakukan pada kasus Onemax dengan panjang individu 20 dan jumlah individu dalam populasi sebesar 50. Masing-masing uji coba dilakukan sebanyak 20 kali dengan iterasi maksimal 1500. Hasil uji coba ini dapat dilihat pada Tabel 1.

Uji coba kedua dilakukan pada kasus Onemax dengan panjang individu 75 dan jumlah individu dalam populasi sebesar 100. Masing-masing uji coba dilakukan sebanyak 20 kali dengan iterasi maksimal 2000. Hasil uji coba ini dapat dilihat pada Tabel 2.

Uji coba ketiga dilakukan pada kasus Onemax dengan panjang individu 100, jumlah individu dalam populasi sebesar 100, dan iterasi maksimal 2000. Masing-masing uji coba dilakukan sebanyak 20 kali dengan. Hasil uji coba ini dapat dilihat pada Tabel 3.

Dari ketiga uji coba tersebut didapatkan parameter GA yang akan digunakan dalam perbandingan pada kasus Onemax, yaitu  $P_c$  50% dan  $P_m$  10%.

**Tabel 1. Tabel Hasil Optimasi Rata-Rata Penyelesaian, Jumlah Iterasi Mencapai Generasi, Waktu Rata-Rata Iterasi Mencapai Generasi Terbaik dan Waktu Total Mencapai Iterasi Maksimal pada Kasus Onemax-20 dengan GA**

Pm	Pc (%)																			
	Hasil Optimasi					Jumlah Iterasi					Waktu Iterasi					Waktu Total				
	5	15	25	35	50	5	15	25	35	50	5	15	25	35	50	5	15	25	35	50
1	20	20	20	20	20	580	386	424	445	319	0,22	0,15	0,18	0,17	0,13	0,57	0,58	0,66	0,58	0,58
10	19,9	19,9	19,9	20	20	503	422	384	469	388	0,23	0,17	0,17	0,19	0,18	0,71	0,63	0,7	0,63	0,72
20	20	19,9	20	20	19,9	546	228	277	452	359	0,22	0,09	0,12	0,18	0,14	0,61	0,62	0,72	0,63	0,62
30	20	20	20	20	20	415	579	489	491	350	0,17	0,24	0,21	0,2	0,14	0,62	0,63	0,62	0,63	0,63
40	19,5	19,7	19,7	19,5	19,7	459	285	323	447	352	0,19	0,12	0,14	0,18	0,14	0,63	0,63	0,66	0,64	0,63
50	19,1	18,9	18,7	19	19	313	413	347	308	311	0,13	0,17	0,17	0,13	0,13	0,62	0,62	0,72	0,63	0,62

**Tabel 2. Tabel Hasil Optimasi Rata-Rata Penyelesaian, Jumlah Iterasi Mencapai Generasi, Waktu Rata-Rata Iterasi Mencapai Generasi Terbaik dan Waktu Total Mencapai Iterasi Maksimal pada Kasus Onemax-75 dengan GA**

Pm	Pc (%)																			
	Hasil Optimasi					Jumlah Iterasi					Waktu Iterasi					Waktu Total				
	5	15	25	35	50	5	15	25	35	50	5	15	25	35	50	5	15	25	35	50
1	63,3	63,5	64,3	64,4	64,6	759	1123	952	980	842	1,33	1,97	1,69	1,76	1,54	3,5	3,52	3,55	3,59	3,66
10	64,4	64,4	64	63,9	64,7	1245	952	1060	935	1150	2,25	1,72	1,94	1,74	2,14	3,6	3,62	3,68	3,72	3,73
20	62,7	63,5	64,4	63,4	63	1226	1252	1024	1275	1081	2,28	2,33	1,93	2,41	2,09	3,71	3,72	3,76	3,78	3,85
30	60,5	62,5	62,1	61,8	62,9	1144	833	1013	926	908	2,16	1,58	1,95	1,79	1,78	3,78	3,79	3,83	3,86	3,92
40	59,2	60,6	61,3	61,4	62	956	855	880	882	943	1,85	1,66	1,74	1,74	1,88	3,85	3,93	3,93	3,93	3,98
50	59,1	59,5	60,7	60,3	59,8	575	529	491	416	415	1,16	1,05	0,97	0,83	0,85	3,94	3,92	3,96	3,98	4,07

**Tabel 3. Tabel Hasil Optimasi Rata-Rata Penyelesaian, Jumlah Iterasi Mencapai Generasi, Waktu Rata-Rata Iterasi Mencapai Generasi Terbaik dan Waktu Total Mencapai Iterasi Maksimal pada Kasus Onemax-100 dengan GA**

Pm	Pc (%)																			
	Hasil Optimasi					Jumlah Iterasi					Waktu Iterasi					Waktu Total				
	5	15	25	35	50	5	15	25	35	50	5	15	25	35	50	5	15	25	35	50
1	79,6	80,7	80,5	81,1	81,5	1037	981	797	979	911	2,37	2,23	1,83	2,27	2,16	4,53	4,55	4,59	4,64	4,72
10	81,1	81,4	81,8	81,9	82	1223	1300	1325	1263	1193	2,85	3,04	3,13	3,03	2,89	4,67	4,68	4,72	4,8	4,84
20	77,8	79	79,8	80,1	81,6	954	963	877	1092	1234	2,3	2,32	2,13	2,67	3,07	4,8	4,81	4,85	4,91	4,96
30	76,2	78,1	78,6	79,3	79,4	903	963	779	1055	973	2,2	2,37	1,94	2,7	2,48	4,88	4,91	4,96	5,07	5,08
40	74,1	76,7	77,5	78,7	79	72	687	977	734	874	1,78	1,72	2,47	1,89	2,27	4,97	5	5,05	5,14	5,19
50	74,1	75,3	77,2	77,5	78,7	721	807	605	814	969	1,82	2,05	1,55	2,11	2,54	5,03	5,06	5,11	5,17	5,22

**Uji Coba Pengaruh Besar Populasi pada Kinerja GA dalam Kasus Onemax**

Uji coba pertama dilakukan pada kasus Onemax dengan panjang individu 20, peluang *crossover* (Pc) sebesar 50%, dan peluang mutasi (Pm) GA sebesar 10%. Masing-masing uji coba dilakukan sebanyak 20 kali dengan iterasi maksimal 1500. Hasil uji coba ini dapat dilihat pada Tabel 4.

**Tabel 4. Tabel Pengaruh Besar Populasi pada Penyelesaian Onemax-20 dengan GA**

Pop	Hasil optimasi	Jml Iterasi	Waktu Iterasi	Waktu Total
30	19,90	529	0,45778	0,15622
40	19,90	210	0,60620	0,12028
50	19,95	388	0,18356	0,72416

Uji coba kedua dilakukan pada kasus Onemax dengan panjang individu 75, peluang *crossover* (Pc) sebesar 50%, dan peluang mutasi (Pm) GA sebesar 10%. Uji coba dilakukan 20 kali dengan iterasi maksimal 1500. Hasil uji coba pada Tabel 5.

**Tabel 5. Tabel Perbandingan Besar Populasi pada Penyelesaian Onemax-75 dengan GA**

Pop	Hasil optimasi	Jml Iterasi	Waktu Iterasi	Waktu Total
100	64,65	1150	2,14370	3,73277
150	65,95	1107	3,04605	5,51246
200	66,95	1187	4,34294	7,33980

#### Uji Coba Pengaruh Parameter BMDA pada Kasus Onemax

Uji coba pertama dilakukan pada kasus Onemax dengan panjang individu 20 dan jumlah individu dalam populasi sebesar 400. Masing-masing uji coba dilakukan sebanyak 20 kali. Hasil uji coba dapat dilihat pada Tabel 6.

**Tabel 6. Tabel Hasil Optimasi Rata-Rata Penyelesaian, Jumlah Iterasi Rata-Rata Penyelesaian, Waktu Konvergensi pada Kasus Onemax-20 dengan BMDA**

New Idv	Jumlah Parents								
	Hasil Optimasi			Jumlah Iterasi			Waktu Konvergensi		
	2	3	4	2	3	4	2	3	4
2	19,95	19,95	17,7	1500	1416	10170	2,8	2,55	14,77
4	19,55	19,8	17,6	763	736	6721	1,49	1,37	10,2

Uji coba kedua dilakukan pada kasus Onemax dengan panjang individu 75 dan jumlah individu dalam populasi sebesar 200. Masing-masing uji coba dilakukan sebanyak 20 kali. Hasil uji coba dapat dilihat pada Tabel 7.

**Tabel 7. Tabel Hasil Optimasi Rata-Rata Penyelesaian, Jumlah Iterasi Rata-Rata Penyelesaian, Waktu Konvergensi pada Kasus Onemax-75 dengan BMDA**

New Idv	Jumlah Parents								
	Hasil Optimasi			Jumlah Iterasi			Waktu Konvergensi		
	2	3	4	2	3	4	2	3	4
2	74,9	74,95	53	1457	1331	20000	29,3	26,2	229,2
4	74,4	74,7	52	803	713	20000	16,9	14,2	207

Dari uji coba ini didapatkan parameter BMDA yang akan digunakan dalam perbandingan pada kasus Onemax, yaitu jumlah *parents* 3 dan banyak individu baru 2.

#### Uji Coba Pengaruh Besar Populasi pada Kinerja BMDA dalam Kasus Onemax

Uji coba pertama dilakukan pada kasus Onemax dengan panjang individu 20, jumlah *parents* 3, dan jumlah individu baru 2. Masing-masing uji coba dilakukan sebanyak 20 kali. Hasil uji coba dapat dilihat pada Tabel 8.

**Tabel 8. Tabel Pengaruh Besar Populasi pada Penyelesaian Onemax-20 dengan BMDA**

Pop	Hasil optimasi	Jumlah Iterasi	Waktu Total
200	19,60	638	1,07496
300	19,95	1102	1,92965
400	19,95	1416	2,54830

Uji coba kedua dilakukan pada kasus Onemax dengan panjang individu 75, jumlah *parents* 3, dan jumlah individu baru 2. Masing-masing uji coba dilakukan sebanyak 20 kali. Hasil uji coba dapat dilihat pada Tabel 9.

**Tabel 9. Tabel Pengaruh Besar Populasi pada Penyelesaian Onemax-75 dengan BMDA**

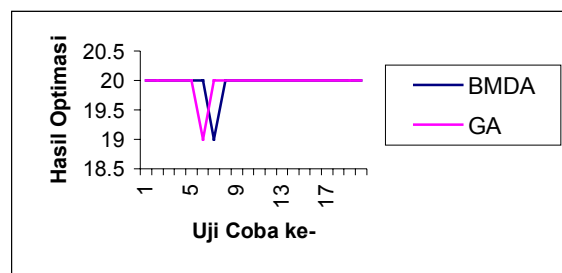
Pop	Hasil optimasi	Jumlah Iterasi	Waktu Total
100	74,85	713	13,43280
150	74,90	992	18,30307
200	74,95	1331	26,17780

#### Perbandingan Kinerja GA dan BMDA pada Kasus Onemax

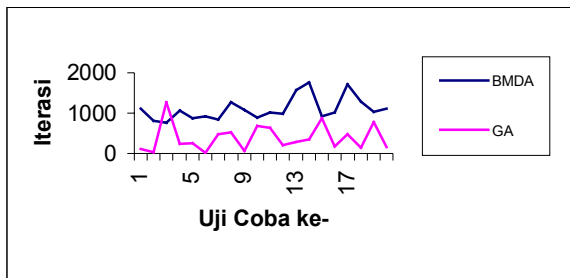
Dalam menyelesaikan kasus Onemax-20, parameter yang digunakan oleh masing-masing algoritma adalah:

- GA → besar populasi 50, Pc = 50%, Pm = 10%, iterasi maksimal 1500
- BMDA → besar populasi 300, jumlah parent = 3, dan jumlah individu baru = 2.

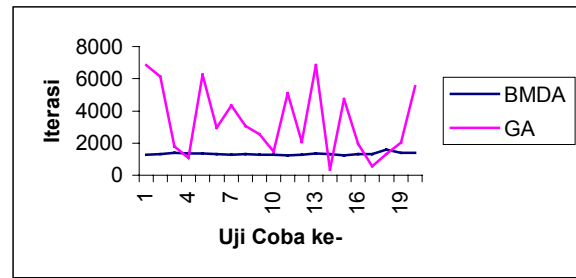
Hasil perbandingan kinerja antara GA dan BMDA pada kasus Onemax-20 dapat dilihat pada Gambar 1 sampai dengan Gambar 3 serta pada Tabel 10.



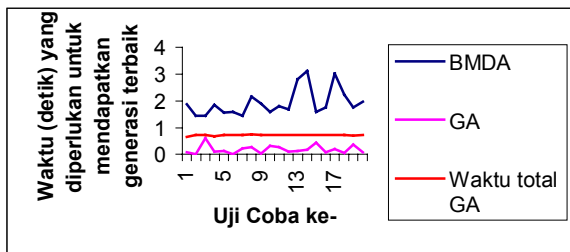
**Gambar 1. Grafik Perbandingan Hasil Optimasi Kasus Onemax-20 antara GA dengan BMDA**



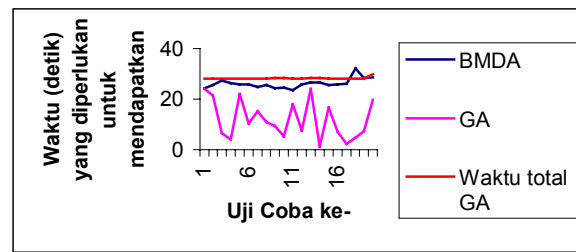
Gambar 2. Grafik Perbandingan Jumlah Iterasi Penyelesaian Kasus Onemax-20 antara GA dengan BMDA



Gambar 5. Grafik Perbandingan Jumlah Iterasi Penyelesaian kasus Onemax-75 antara GA dengan BMDA



Gambar 3. Grafik Perbandingan Waktu Penyelesaian Kasus Onemax-20 antara GA dengan BMDA



Gambar 6. Grafik Perbandingan Waktu Penyelesaian Kasus Onemax-75 antara GA dengan BMDA

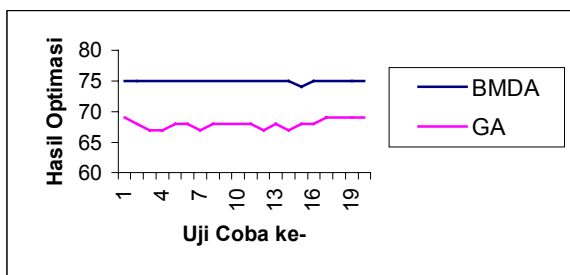
Tabel 10. Tabel Perbandingan Penyelesaian Kasus Onemax-20 antara GA dengan BMDA.

Algoritma	Hasil optimasi	Jml Iterasi	Waktu Iterasi	Waktu Total
GA	19,95	388	0,18356	0,72416
BMDA	19,95	1102	1,92965	

Dalam menyelesaikan kasus Onemax-75, parameter yang digunakan oleh masing-masing algoritma adalah:

- GA → besar populasi 200,  $P_c = 50\%$ ,  $P_m = 10\%$ , iterasi maksimal 8000
- BMDA → besar populasi 200, jumlah *parent* = 3, dan jumlah individu baru = 2.

Hasil perbandingan kinerja antara GA dan BMDA pada kasus Onemax-75 dapat dilihat pada Gambar 4 sampai dengan Gambar 6 serta pada Tabel 11.



Gambar 4. Grafik Perbandingan Hasil Optimasi Kasus Onemax-75 antara GA dengan BMDA

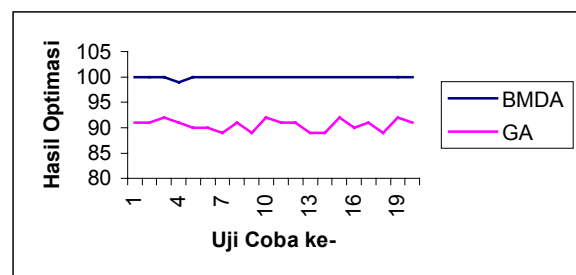
Tabel 11. Tabel Perbandingan Penyelesaian Kasus Onemax-75 antara GA dengan BMDA

Algoritma	Hasil optimasi	Jumlah Iterasi	Waktu Iterasi	Waktu Total
GA	68,00	3362	12,01559	28,28278
BMDA	74,95	1331	26,17780	

Dalam menyelesaikan kasus Onemax-100, parameter yang digunakan oleh masing-masing algoritma adalah:

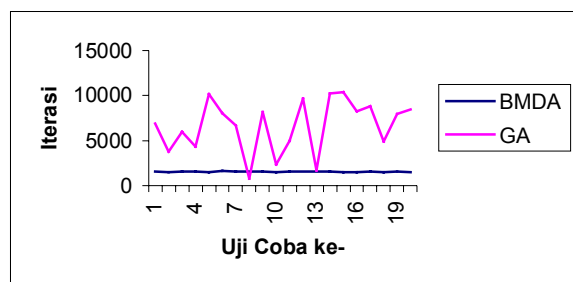
- GA → besar populasi 200,  $P_c = 50\%$ ,  $P_m = 10\%$ , iterasi maksimal 11000
- BMDA → besar populasi 200, jumlah *parent* = 3, jumlah individu baru = 2.

Hasil perbandingan kinerja antara GA dan BMDA pada kasus Onemax-100 dapat dilihat pada Gambar 7 sampai dengan Gambar 9 serta pada Tabel 12.

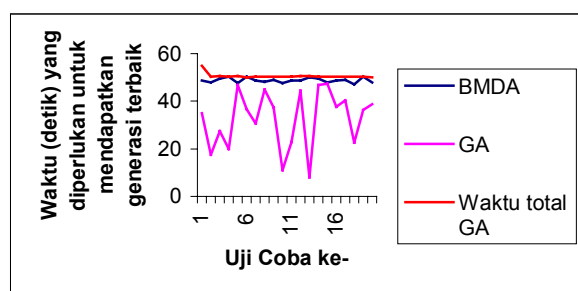


Gambar 7. Grafik Perbandingan Hasil Optimasi Kasus Onemax-100 antara GA dengan BMDA





**Gambar 8. Grafik Perbandingan Jumlah Iterasi Penyelesaian Kasus Onemax-100 antara GA dengan BMDA**



**Gambar 9. Grafik Perbandingan Waktu Penyelesaian Kasus Onemax-100 antara GA dengan BMDA**

**Tabel 12. Tabel Perbandingan Penyelesaian Kasus Onemax-100 antara GA dengan BMDA**

	Hasil optimasi	Jumlah Iterasi	Waktu Iterasi	Waktu Total
GA	90,55	7075	32,65856	50,53747
BMDA	99,95	1552	48,77570	

Dari perbandingan di atas, dapat diketahui bahwa GA unggul dalam hal jumlah iterasi dan waktu proses pada kasus Onemax dengan ukuran kecil (Onemax-20). Namun pada kasus Onemax dengan ukuran yang lebih besar (Onemax-75 dan Onemax-100), BMDA menunjukkan keunggulan dalam hal kemampuan

mencapai hasil yang lebih optimal dan jumlah iterasi yang lebih kecil pada kisaran waktu yang sama.

### Uji Coba Pengaruh Parameter GA pada Fungsi De Jong F2

Uji coba dilakukan pada fungsi De Jong F2 dengan jumlah individu dalam populasi sebesar 50. Masing-masing uji coba dilakukan sebanyak 20 kali dengan iterasi maksimal 2000. Hasil uji coba dapat dilihat pada Tabel 13.

Dari uji coba ini didapatkan parameter GA yang akan digunakan dalam perbandingan pada Fungsi De Jong F2, yaitu Pc 50% dan Pm 20%.

### Uji Coba Pengaruh Besar Populasi pada Kinerja GA dalam Fungsi De Jong F2

Uji coba dilakukan pada fungsi De Jong F2 dengan peluang *crossover* (Pc) sebesar 50% dan peluang mutasi (Pm) GA sebesar 20%. Masing-masing uji coba dilakukan sebanyak 20 kali dengan iterasi maksimal 2000. Hasil uji coba dapat dilihat pada Tabel 14.

**Tabel 14. Tabel Pengaruh Besar Populasi pada Penyelesaian Fungsi DeJong F2 dengan GA**

Pop	Hasil optimasi	Jumlah Iterasi	Waktu Iterasi	Waktu Total
30	0,00035	934	0,41637	0,90465
40	0,00025	574	0,31716	1,11324
50	0,00000	675	0,44059	1,31482

### Uji Coba Pengaruh Parameter BMDA pada Fungsi De Jong F2

Uji coba dilakukan pada fungsi De Jong F2 dengan jumlah individu dalam populasi sebesar 400. Masing-masing uji coba dilakukan sebanyak 20 kali. Hasil uji coba dapat dilihat pada Tabel 15.

**Tabel 13. Tabel Hasil Optimasi Rata-Rata, Jumlah Iterasi, Waktu Iterasi dan Waktu Total pada Penyelesaian Fungsi DeJong F2 dengan GA**

Pm	Pc (%)																			
	Hasil Optimasi					Jumlah Iterasi					Waktu Iterasi					Waktu Total				
	5	15	25	35	50	5	15	25	35	50	5	15	25	35	50	5	15	25	35	50
1	0,03	0,01	0,01	0,01	0,01	878	752	1041	680	708	0,55	0,47	0,76	0,44	0,44	1,26	1,26	1,41	1,27	1,26
10	0	0	0	0	0	695	640	678	626	849	0,46	0,42	0,45	0,41	0,59	1,32	1,31	1,34	1,31	1,38
20	0	0	0	0	0	669	621	619	555	675	0,43	0,4	0,42	0,36	0,44	1,29	1,3	1,31	1,31	1,31
30	0,01	0	0,01	0	0	583	374	328	763	753	0,37	0,25	0,23	0,5	0,49	1,3	1,32	1,34	1,32	1,32
40	0,06	0,08	0,07	0,05	0,06	452	510	332	388	404	0,3	0,33	0,22	0,26	0,27	1,31	1,31	1,34	1,33	1,33
50	0,1	0,25	0,21	0,15	0,36	270	204	166	282	255	0,18	0,14	0,11	0,19	0,17	1,32	1,33	1,33	1,34	1,34

**Tabel 15. Tabel Hasil Optimasi Rata-Rata, Jumlah Iterasi, Waktu Konvergensi pada Penyelesaian Fungsi DeJong F2 dengan BMDA**

New Idv	Jumlah Parents								
	Hasil Optimasi			Jumlah Iterasi			Waktu Konvergensi		
	2	3	4	2	3	4	2	3	4
2	0,003	0,003	0,23	6280	7561	40000	23,85	26,95	142
4	0,004	0,003	0,01	2432	5934	40000	9,362	21,94	141

Dari uji coba ini didapatkan parameter BMDA yang akan digunakan dalam perbandingan pada Fungsi De Jong F2, yaitu jumlah parents 3 dan banyak individu baru 2.

#### Uji Coba Pengaruh Besar Populasi pada Kinerja BMDA dalam Fungsi De Jong F2

Uji coba dilakukan pada fungsi De Jong F2 dengan jumlah *parents* sebanyak 3 dan jumlah individu baru tiap iterasi sebanyak 2. Masing-masing uji coba dilakukan sebanyak 20 kali. Hasil uji coba dapat dilihat pada Tabel 16.

**Tabel 16. Tabel Pengaruh Besar Populasi pada Penyelesaian Fungsi DeJong F2 dengan BMDA**

Pop	Hasil optimasi	Jumlah Iterasi	Waktu Total
50	0,51190	611	1,37030
100	0,07550	1781	4,42490
400	0,00250	7561	26,94840

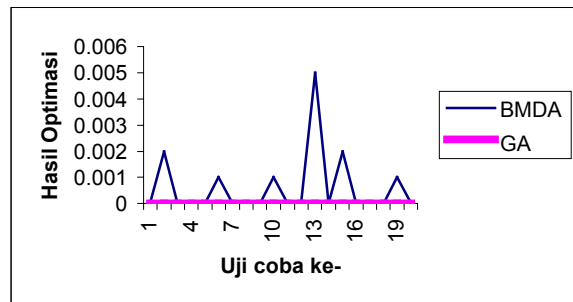
#### Perbandingan Kinerja GA dan BMDA pada Fungsi De Jong F2

Dalam menyelesaikan Fungsi De Jong F2 ini, parameter yang digunakan oleh masing-masing algoritma adalah:

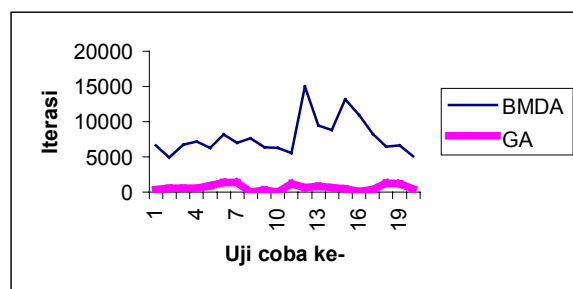
- GA → besar populasi 50,  $P_c = 50\%$ ,  $P_m = 20\%$ , iterasi maksimal 2000
- BMDA → besar populasi 400, jumlah *parent* = 3, dan jumlah individu baru = 2.

Hasil perbandingan ini dapat dilihat pada Gambar 10 sampai dengan Gambar 12 serta Tabel 17.

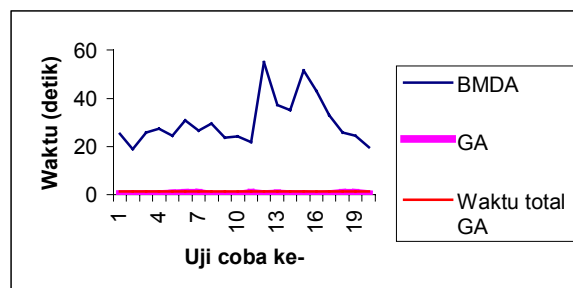
Dari perbandingan tersebut, dapat diketahui bahwa GA menunjukkan kinerja yang lebih baik daripada BMDA pada Fungsi De Jong F2 utamanya dalam hal jumlah iterasi dan waktu proses.



**Gambar 10. Grafik Perbandingan Hasil Optimasi Fungsi De Jong F2 antara GA dengan BMDA**



**Gambar 11. Grafik Perbandingan Jumlah Iterasi Penyelesaian Fungsi De Jong F2 antara GA dengan BMDA**



**Gambar 12. Grafik Perbandingan Waktu Penyelesaian Fungsi De Jong F2 antara GA dengan BMDA**

**Tabel 17. Tabel Perbandingan Penyelesaian Fungsi De Jong F2 antara GA dengan BMDA**

	Hasil optimasi	Jumlah Iterasi	Waktu Iterasi	Waktu Total
GA	0,00000	675	0,44059	1,31482
BMDA	0,00250	7561	26,94840	



**Tabel 18. Tabel Hasil Optimasi Rata-Rata, Jumlah Iterasi, Waktu Iterasi, Waktu Total pada Penyelesaian TSP-Rute 1 dengan GA**

Pm	Pc (%)																			
	Hasil Optimasi					Jumlah Iterasi					Waktu Iterasi					Waktu Total				
	5	15	25	35	50	5	15	25	35	50	5	15	25	35	50	5	15	25	35	50
1	1958	1822	1637	1595	1469	978	1142	1392	1207	1365	5,37	6,28	7,68	6,71	7,69	10,81	10,84	10,9	11,19	11,11
10	1969	1883	1800	1813	1695	1526	1359	1531	1470	1465	8,54	7,65	8,62	8,4	8,46	11,12	11,15	11,22	11,34	11,46
20	2538	2426	2265	2213	2127	1268	1261	1003	761	1303	7,26	7,27	5,83	4,46	7,75	11,33	11,4	11,46	11,64	11,75
30	3008	2454	2688	2270	2390	1232	628	788	1378	571	7,18	3,72	4,69	8,2	3,53	11,52	11,64	11,68	11,84	12,07
40	3011	2710	2459	2525	2458	410	464	753	464	396	2,49	2,82	4,5	2,86	2,57	11,8	11,8	11,73	11,92	12,25
50	2923	2713	2683	2595	2579	586	311	302	289	399	3,52	1,89	1,86	1,79	2,44	11,83	11,87	11,97	12	12,08

**Tabel 19. Tabel Hasil Optimasi Rata-Rata, Jumlah Iterasi, Waktu Iterasi, Waktu Total pada Penyelesaian TSP-Rute 2 dengan GA**

Pm	Pc (%)																			
	Hasil Optimasi					Jumlah Iterasi					Waktu Iterasi					Waktu Total				
	5	15	25	35	50	5	15	25	35	50	5	15	25	35	50	5	15	25	35	50
1	1731	1733	1701	1685	1611	1534	808	1560	1236	1255	8,18	4,39	8,42	6,74	6,92	10,6	10,65	10,73	10,78	10,87
10	1999	2011	1876	1805	1700	1520	1552	1550	1456	1219	8,44	8,63	8,65	8,16	7	10,97	11,03	11,08	11,1	11,33
20	2348	2185	2157	2127	2079	1389	1260	1338	1147	1398	7,82	7,21	7,62	6,58	8,05	11,17	11,3	11,29	11,34	11,44
30	2462	2342	2269	2284	2184	1173	1413	952	1001	996	6,78	8,04	5,56	5,89	5,9	11,39	11,31	11,48	11,64	11,62
40	2374	2485	2287	2413	2316	1041	1133	740	218	705	6,18	6,62	4,38	1,33	4,24	11,73	11,57	11,71	11,7	11,77
50	2618	2614	2484	2423	2382	654	485	348	457	230	3,87	2,84	2,09	2,79	1,43	11,57	11,65	11,72	11,73	11,82

**Uji Coba Pengaruh Parameter GA pada Kasus TSP**

Uji coba dilakukan pada kasus TSP-Rute 1 dengan jumlah individu dalam populasi sebesar 100. Masing-masing uji coba dilakukan sebanyak 10 kali dengan iterasi maksimal 2000. Hasil uji coba dapat dilihat pada Tabel 18.

Uji coba dilakukan pada kasus TSP-Rute 2 dengan jumlah individu dalam populasi sebesar 300. Masing-masing uji coba dilakukan sebanyak 10 kali dengan iterasi maksimal 2000. Hasil uji coba dapat dilihat pada Tabel 19.

Dari uji coba ini didapatkan parameter GA yang akan digunakan dalam perbandingan pada kasus TSP, yaitu Pc 50% dan Pm 1%.

**Uji Coba Pengaruh Besar Populasi pada Kinerja GA dalam Kasus TSP**

Uji coba pertama dilakukan pada kasus TSP-Rute 1 dengan peluang *crossover* (Pc) sebesar 50% dan peluang mutasi (Pm) GA sebesar 1%. Masing-masing uji coba dilakukan sebanyak 10 kali dengan iterasi maksimal 2000. Hasil uji coba dapat dilihat pada Tabel 20.

**Tabel 20. Tabel Pengaruh Besar Populasi pada Penyelesaian TSP-Rute 1 dengan GA.**

Pop	Hasil optimasi	Jumlah Iterasi	Waktu Iterasi	Waktu Total
100	1469,00	1365	7,68747	11,11089
200	1317,90	1419	15,52340	21,74682
300	1180,20	1212	19,81870	32,47184

Uji coba kedua dilakukan pada kasus TSP-Rute 2 dengan peluang *crossover* (Pc) sebesar 50% dan peluang mutasi (Pm) GA sebesar 1%. Masing-masing uji coba dilakukan sebanyak 10 kali dengan iterasi maksimal 2000. Hasil uji coba dapat dilihat pada Tabel 21.

**Tabel 21. Tabel Pengaruh Besar Populasi pada Penyelesaian TSP-Rute 2 dengan GA**

Pop	Hasil optimasi	Jumlah Iterasi	Waktu Iterasi	Waktu Total
100	1611,00	1255	6,92184	10,87497
200	1512,70	1047	11,47809	21,42185
300	1500,80	1363	21,56090	31,45465

### Uji Coba Pengaruh Parameter BMDA pada Kasus TSP

Uji coba pertama dilakukan pada pada kasus TSP-Rute 1 dengan jumlah individu dalam populasi sebesar 300. Masing-masing uji coba dilakukan sebanyak 10 kali. Hasil uji coba dapat dilihat pada Tabel 22.

Uji coba kedua dilakukan pada pada kasus TSP-Rute 2 dengan jumlah individu dalam populasi sebesar 300. Masing-masing uji coba dilakukan sebanyak 10 kali. Hasil uji coba dapat dilihat pada Tabel 23.

Dari kedua uji coba ini didapatkan parameter BMDA yang akan digunakan pada kasus TSP, yaitu jumlah parents 3 dan banyak individu baru 2.

**Tabel 22. Tabel hasil optimasi rata-rata, jumlah iterasi, waktu konvergensi pada penyelesaian TSP-Rute 1 dengan BMDA.**

New Idv	Jumlah Parents								
	Hasil Optimasi			Jumlah Iterasi			Waktu Konvergensi		
	2	3	4	2	3	4	2	3	4
2	995,6	936	936	1906	668	277	161,52	136,88	20,273
4	1511,6	1126,2	936	1017	949	143	87,741	79,55	10,941

**Tabel 23. Tabel Hasil Optimasi Rata-Rata, Jumlah Iterasi, Waktu Konvergensi pada Penyelesaian TSP-Rute 2 dengan BMDA**

New Idv	Jumlah Parents								
	Hasil Optimasi			Jumlah Iterasi			Waktu Konvergensi		
	2	3	4	2	3	4	2	3	4
2	1540,1	1427,6	2143,4	3823	4954	2645	317,32	365,6	215,76
4	2068,8	1615	2193,4	1316	2118	705	109,16	160,88	58,475

### Uji Coba Pengaruh Besar Populasi pada Kinerja BMDA dalam Kasus TSP

Uji coba pertama dilakukan pada pada kasus TSP-Rute 1 dengan jumlah *parents* sebanyak 3 dan jumlah individu baru tiap iterasi sebanyak 2. Masing-masing uji coba dilakukan sebanyak 10 kali. Hasil uji coba dapat dilihat pada Tabel 24.

Uji coba kedua dilakukan pada pada kasus TSP-Rute 2 dengan jumlah *parents* sebanyak 3 dan jumlah individu baru tiap iterasi sebanyak 2. Masing-masing uji coba dilakukan sebanyak 10 kali. Hasil uji coba dapat dilihat pada Tabel 25.

**Tabel 24. Tabel Pengaruh Besar Populasi pada Penyelesaian TSP-Rute 1 dengan BMDA**

Pop	Hasil optimasi	Jumlah Iterasi	Waktu Total
100	1528,50	669	51,78278
200	1020,00	1228	97,57475
300	936,00	1683	136,87650

**Tabel 25. Tabel Pengaruh Besar Populasi pada Penyelesaian TSP-Rute 2 dengan BMDA**

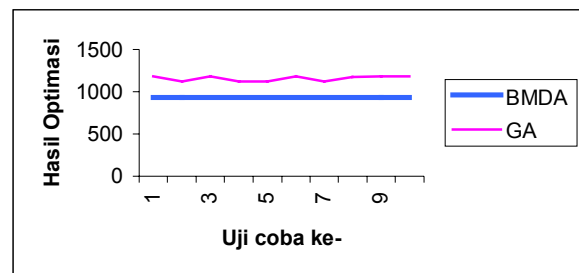
Pop	Hasil optimasi	Jumlah Iterasi	Waktu Total
300	1427,60	4954	365,59530
400	1386,00	7577	606,45780

### Perbandingan Kinerja GA dan BMDA pada Kasus TSP

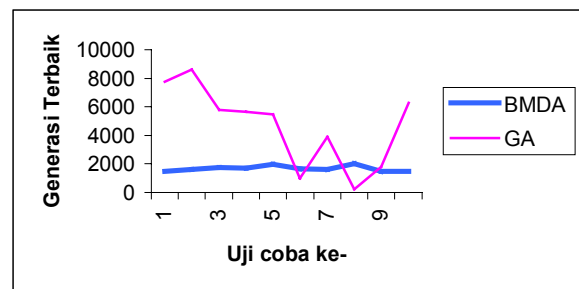
Dalam menyelesaikan kasus TSP-Rute 1, parameter yang digunakan oleh masing-masing algoritma adalah:

- GA → besar populasi 300,  $P_c = 50\%$ ,  $P_m = 1\%$ , iterasi maksimal 9000
- BMDA → besar populasi 300, jumlah parent = 3, dan jumlah individu baru = 2.

Perbandingan kinerja GA dengan BMDA dalam TSP-Rute 1 dapat dilihat pada Gambar 13 sampai dengan Gambar 15 serta Tabel 26.



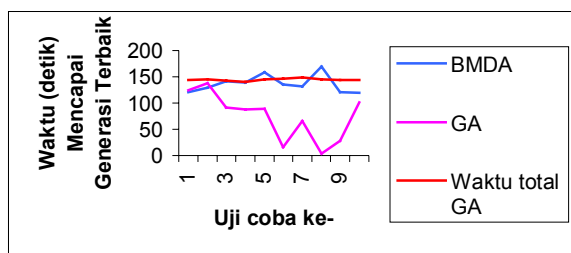
**Gambar 13. Grafik Perbandingan Hasil Optimasi Kasus TSP-Rute 1 antara GA dengan BMDA**



**Gambar 14. Grafik Perbandingan Jumlah Iterasi Penyelesaian Kasus TSP-Rute 1 antara GA dengan BMDA**

**Tabel 26. Tabel Perbandingan Penyelesaian Kasus TSP-Rute 1 antara GA dengan BMDA**

Algoritma	Hasil optimasi	Jumlah Iterasi	Waktu Iterasi	Waktu Total
GA	1155,40	4648	74,97650	144,92340
BMDA	936,00	1683		136,87650



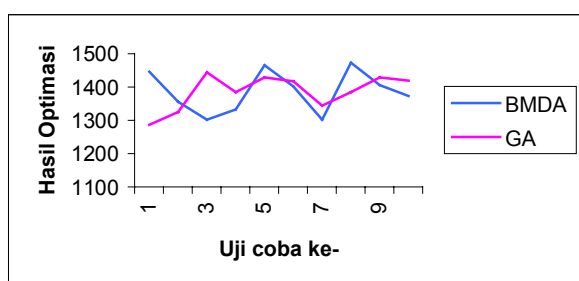
**Gambar 15. Grafik Perbandingan Waktu Penyelesaian Kasus TSP-Rute 1 antara GA dengan BMDA**

Dalam menyelesaikan kasus TSP-Rute 2, parameter yang digunakan oleh masing-masing algoritma adalah:

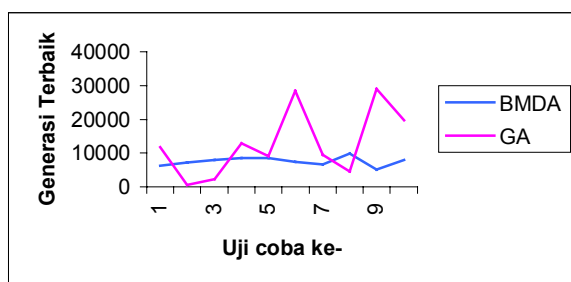
- GA → besar populasi 400,  $P_c = 50\%$ ,  $P_m = 1\%$ , iterasi maksimal 30000
- BMDA → besar populasi 400, jumlah *parent* = 3, dan jumlah individu baru = 2.

Perbandingan kinerja GA dengan BMDA dalam TSP-Rute 2 dapat dilihat pada Gambar 16 sampai dengan Gambar 18 serta Tabel 27.

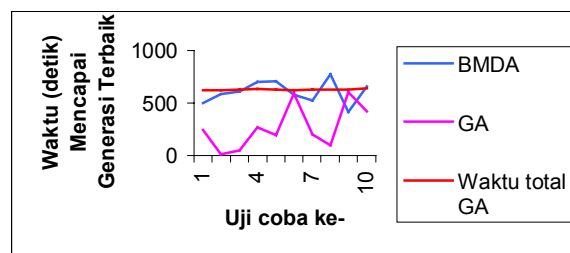
Dari perbandingan tersebut, dapat diketahui bahwa dalam menyelesaikan kasus TSP, BMDA menunjukkan kinerja yang lebih baik dalam hal hasil optimasi dan jumlah iterasi dalam kisaran waktu proses yang sama.



**Gambar 16. Grafik Perbandingan Hasil Optimasi Kasus TSP-Rute 2 antara GA dengan BMDA**



**Gambar 17. Grafik Perbandingan Jumlah Iterasi Penyelesaian Kasus TSP-Rute 2 antara GA dengan BMDA**



**Gambar 18. Grafik Perbandingan Waktu Penyelesaian Kasus TSP-Rute 2 antara GA dengan BMDA**

**Tabel 27. Tabel Perbandingan Penyelesaian Kasus TSP-Rute 2 antara GA dengan BMDA**

Algoritma	Hasil optimasi	Jumlah Iterasi	Waktu Iterasi	Waktu Total
GA	1386,40	12796	268,14059	628,09215
BMDA	1386,00	7577		606,45780

## PENUTUP

Beberapa kesimpulan yang dapat ditarik dari hasil pengamatan selama proses perancangan, pembuatan sampai dengan uji coba Penelitian ini adalah sebagai berikut:

1. Parameter pada Algoritma Genetika (GA) adalah peluang *crossover* ( $P_c$ ) dan peluang mutasi ( $P_m$ ), sedangkan parameter Algoritma *Bivariate Marginal Distribution* (BMDA) adalah jumlah *parents* dan banyak individu baru. Berikut ini pengaruh parameter dan jumlah populasi pada masing-masing algoritma:
  - Secara umum, baik pada GA maupun BMDA, semakin besar populasi yang digunakan, hasil optimasi akan cenderung bertambah baik, namun waktu proses akan bertambah lama. BMDA cenderung membutuhkan populasi yang lebih besar dari GA (khususnya pada masalah dengan ukuran kecil) karena membutuhkan keragaman individu yang lebih besar untuk membentuk individu baru yang lebih baik sebelum populasi menjadi konvergen.
  - Untuk Algoritma Genetika:
    - Makin besar nilai  $P_c$  dan nilai  $P_m$ , makin besar pula waktu total untuk mencapai iterasi maksimal.
    - Nilai peluang mutasi ( $P_m$ ) yang terlalu besar akan memberikan hasil yang buruk karena gangguan terhadap populasi menjadi terlalu besar. Nilai  $P_m$  yang cenderung memberikan hasil optimasi terbaik adalah sebesar 1% dan 10%.

- Untuk Algoritma *Bivariate Marginal Distribution*:
  - Jumlah *parents* yang terlalu banyak akan membuat algoritma sulit untuk membuat individu baru yang baik. Jumlah *parents* yang memiliki kecenderungan untuk memberikan hasil yang baik adalah sebanyak 3.
  - Jumlah individu baru yang dapat memberikan hasil yang baik adalah sebanyak 2. Jumlah individu baru yang banyak akan membuat populasi menjadi lebih cepat menjadi konvergen yang menyebabkan algoritma berhenti saat hasil yang didapatkan mungkin belum terlalu baik.
- 2. Untuk penyelesaian kasus Onemax dengan ukuran masalah yang tidak terlalu besar (ditunjukkan oleh percobaan Onemax-20), GA lebih unggul dalam hal jumlah iterasi yang lebih kecil dan waktu yang lebih cepat dibanding BMDA. Namun, untuk penyelesaian kasus Onemax dengan ukuran masalah yang cukup besar (ditunjukkan oleh percobaan Onemax-100), BMDA lebih mampu memberikan hasil optimasi dan jumlah iterasi yang lebih baik dibanding GA dalam kisaran waktu proses yang sama. Parameter GA yang cocok untuk permasalahan Onemax adalah peluang *crossover* (Pc) 50% dan peluang mutasi (Pm) 10%, sedangkan parameter BMDA adalah jumlah *parents* 3 dan jumlah individu baru 2.
- 3. Untuk kasus Fungsi De Jong F2, GA menunjukkan kinerja yang lebih baik dibanding BMDA. Dalam hal hasil optimasi, GA cenderung selalu memberikan hasil yang lebih baik. Dalam hal jumlah iterasi dan waktu yang dibutuhkan, GA juga lebih unggul. Parameter GA yang cocok untuk permasalahan Onemax adalah peluang *crossover* (Pc) 50% dan peluang mutasi (Pm) 20%, sedangkan parameter BMDA adalah jumlah *parents* 3 dan jumlah individu baru 2.
- 4. Untuk kasus *Traveling Salesman Problem*, BMDA menunjukkan kinerja yang lebih baik dalam hal hasil optimasi dan jumlah iterasi dalam kisaran waktu proses yang sama. Parameter GA yang cocok untuk permasalahan Onemax adalah peluang *crossover* (Pc) 50% dan peluang mutasi (Pm) 1%, sedangkan parameter BMDA adalah jumlah *parents* 3 dan jumlah individu baru 2.
- 5. Secara umum, untuk masalah dengan ukuran panjang individu yang kecil (pada penyelesaian kasus Onemax-20 dan Fungsi De Jong F2), GA lebih baik dari BMDA dalam hal jumlah iterasi dan waktu proses yang lebih cepat. Sedangkan untuk masalah dengan ukuran panjang individu yang besar (pada penyelesaian kasus Onemax-100

dan TSP), BMDA lebih unggul dalam hal hasil optimasi dan jumlah iterasi dalam kisaran waktu proses yang sama.

## DAFTAR PUSTAKA

1. Kumar, Paul Topon & Hitoshi, Iba. *Linear and Combinatorial Optimizations by Estimation of Distribution Algorithms*. 9<sup>th</sup> MPS symposium on Evolutionary Computation, IPSJ, Japan, 2002.
2. Kusumadewi, Sri. *Artificial Intelligence (Teknik dan Aplikasinya)*. Penerbit Graha Ilmu. Yogyakarta, 2003.
3. Michalewicz, Zbigniew. *Genetic Algorithms + Data Structures = Evolution Programs*, Third, Revised and Extended Edition. Springer-Verlag. Berlin, 1999.
4. Pelikan, Martin & Muehlenbein, Heinz. *The Bivariate Marginal Distribution Algorithm*. Dept. of Mathematics, Slovak Technical University. Bratislava, Slovakia, 1999.